A Markovian Model For Contour Grouping

Sabine Urago, Josiane Zerubia, Marc Berthod

▶ **To cite this version:**

Sabine Urago, Josiane Zerubia, Marc Berthod. A Markovian Model For Contour Grouping. [Research Report] RR-2122, INRIA. 1994. inria-00074550

HAL Id: inria-00074550
https://inria.hal.science/inria-00074550

Submitted on 24 May 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET AUTOMATIQUE

A Markovian Model For Contour Grouping.

Sabine URAGO , Josiane ZERUBIA , Marc BERTHOD

N° 2122

1994

PROGRAMME 4

Robotique,

image

et vision

1994

ISSN 0249-6399

INRIA
SOPHIA ANTIPOLIS

A Markovian Model For Contour Grouping.

Sabine URAGO , Josiane ZERUBIA , Marc BERTHOD

Programme 4 — Robotique, image et vision
Projet PASTIS

Rapport de recherche n ° 2122 — 1994 — 27 pages

Abstract: In order to interpret and analyse a scene, determining the contours is a fundamental step.
Classical methods of contour extraction do not always allow the detection of all the contours. We notice, for example, that the contours obtained by a Canny-Deriche filter have some gaps, especially at corners or at T-junctions. In short, the boundaries which are detected are not always closed.
In this report, we present an algorithm that restores incomplete contours.
We model the image by Markov Random Fields and we define the Gibbs Distribution associated with it. In order to complete the contours, several criteria are defined and introduced in an energy function, which has to be optimized. The deterministic ICM ("Iterated Conditional Mode") relaxation algorithm is implemented to minimize this energy function. The result is a contour image consisting of closed contours.
This method has been tested on different images which present different types of difficulties (indoors, outdoors, satellite (SPOT), industrial and medical images).

Key-words: Contour detection, Markov Random Fields, Gibbs distributions, relaxation algorithms.

(Résumé : tsvp)

ACKNOWLEDGMENTS: We would like to thank the GdR TdSI (CNRS) for providing part of the images presented in this report.

Une Modélisation Markovienne pour groupements de contours.

Résumé : Pour interpréter et analyser une scène, la détermination des contours est une étape fondamentale.
Les méthodes classiques de détection de contours, ne permettent pas toujours la détection de tous les contours. En effet, les contours obtenus par exemple, par un filtrage de Canny-Deriche suivi des suppressions des non-maxima, sont souvent corrects mais contiennent parfois des discontinuités (notamment aux niveaux des "coins" et des "jonctions en T"). Les contours détectés ne sont donc pas systématiquement fermés.
Dans ce rapport, nous présentons un algorithme permettant de restaurer des images de contours incomplets.
Nous modélisons l'image par des champs de Markov et définissons les distributions de Gibbs qui leurs sont associées. Pour cela, certains critères à optimiser sont déterminés afin de complèter les parties manquantes. La mise en oeuvre d'une méthode de relaxation déterministe ICM ("Iterated Conditional Mode") permet d'aboutir à une configuration dans laquelle les contours sont complétés.
Cette méthode a été testée sur différentes images présentant diverses difficultés (images de scènes réelles d'intérieur, d'extérieur, satellite (SPOT), médicales).

Mots-clé : Détection de contours, champs de Markov, distributions de Gibbs, algorithmes de relaxation.

Contents

List of Figures

List of Tables

1 Introduction.

In this report, we describe an algorithm that restores images of incomplete contours.

Classical methods of contour detection do not always allow the detection of all the contours [3], [9], [19]. The contours obtained by a Canny-Deriche filter for example [4], [5], sometimes have some interruptions (especially at corners or at T-junctions).

Contour detection has been the object of many works. Among these works, we can quote the one of Shaashua and Ullman [14] which introduces a new grouping method based on a locally connected network. Marroquin [12], [13] restores incomplete contour images with a Markovian model, but he only performs the restoration of synthetic images. In [16] and [17], we have modified and extended his method, in order to restore real images. We have experimented this algorithm on satellite images (SPOT). Tan, Gelfand and Delp [15] detecte edges by finding the edge configuration that minimize a cost function which is optimized by simulated annealing [7]. Other authors have also presented contour detection methods using two deterministic relaxation algorithms : GNC ("Gratuated Non convexity") [3], and Mean Field Annealing [6], [18].

In this report, we use a Markovian model [1], [2], [7], to restore the contour image. This means that the state of each site depends only on the state of its neighboring sites, and not on the state of all sites in the image. An interesting property is that the state of one site can specify a local information, but indirectly also a more global one.

We will describe the Gibbs distribution associated with the proposed model. We will then define some criteria which are introduced into the energy function to be minimized. A deterministic relaxation algorithm ICM ("Iterated Conditional Mode") [2] is used in order to generate an optimal configuration, in which the contours are reconstructed.

To illustrate this algorithm, several examples of real image restorations have been tested. Those examples present different types of difficulties (indoors, outdoors, satellite (SPOT), industrial and medical images).

2 Description of the Markovian model.

We first describe the selected Markov Random Field (MRF) model :

At each site, that is to say at each pixel of the image, we use 8-connexity to determine the MRF.

At a given site $\mathbf{i}$, we store three components. The first and the second components of the MRF correspond to global information. The first specifies the number of the

contour chain, and the second its global direction. The third one determines a local component : the state.

Notations :
Given E_i, the MRF at the site **i** is defined as : $E_i = (ch_i, D_{\theta_i}, st_i)$

where

- ch_i : **the number of the chain** (i.e. the contour running through the pixel).
- D_{θ_i} : **the direction**.
- st_i : **the state**.

Now, we briefly give a definition of those three quantities (more details will be given afterwards).

2.1 First component : the number of the chain.

The contours are decomposed into straight line segments (chains), which are defined as a set of pixels approximately aligned. Each chain is identified by a number. The first component of the MRF corresponds to the number of the chain.

2.2 Second component : the direction.

The second component, **the direction** (written $\mathbf{D}_\theta$), is a **vector with four real components**.

This vector is defined in order to have information about the average contour line direction.

$$D_\theta = (D_{\theta_0}, D_{\theta_1}, D_{\theta_2}, D_{\theta_3})$$

where

- $\theta_\mathbf{k}$ is the k^{th} predefined direction which makes an angle of $(k\Pi/4)$ with the horizontal ($\mathbf{k} \in [0..3]$, see Fig. 1).
- $\mathbf{D}_{\theta_\mathbf{k}}$ is a likelihood measure of the direction $\theta_\mathbf{k}$.

The four directions θ_k are represented in the following figure :

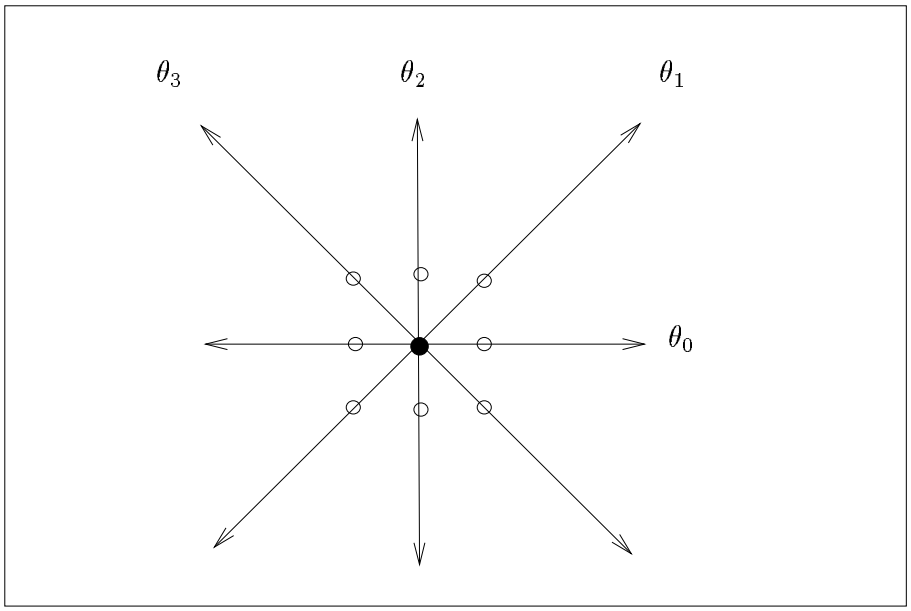

Figure 1: Representation of the four directions.

2.3 Third component : the state.

We define **four possible states** :

- **No contour** (written **"O"**).
- **Termination** (**"T"**) : A **Termination** corresponds to the end of the contour chain.
- **Corner** (**"C"**) : A **Corner** represents a "T-junction" or a "corner". It is in fact a pixel that connects different chains of contours.
- **Straight continuation** (**"D"**) : A **Straight continuation** is associated to the other pixels of a chain.

This model is interesting because it allows us to have, at each site, not only local information (state), but also more global information (chain and direction).

3 Initialization of the Markov Random Field.

To define the initial contour image **I**, we use a classical method : the Canny-Deriche filter with non-maxima suppression [4], [5].

Hence, at each point, the image **I** indicates the norm of the gradient.

$$I(s) = \begin{cases} 0 & \text{if the site } \mathbf{s} \text{ does not belong to a contour} \\ \|\vec{\bigtriangledown}(s)\| & \text{otherwise} \end{cases}$$

To initialize the three components of the **MRF**, we look at the initial data given by the image **I**.

3.1 Initialization of the chain number.

Three steps enable the initialization of this component :

- Determination for each site of the "best" direction.
- Creation of the contour chains.
- Suppression of small chains .

3.1.1 First step : Determination of the "best" direction.

The first step consists of the determination of the "best" direction $\theta_{\mathbf{max}}$, at each site where a contour has been detected.

$\theta_{\mathbf{max}}$ is selected among the four possible directions ($\theta_0, \theta_1, \theta_2, \theta_3$).

For this selection, we apply successively on the initial image **I**, four filters $\mathbf{M}_{\theta_\mathbf{k}}$ associated respectively with the directions $\theta_\mathbf{k}$.

For example, the filter $\mathbf{M_{\theta_1}}$ is represented in the following figure (for more details(see [10], [11])):

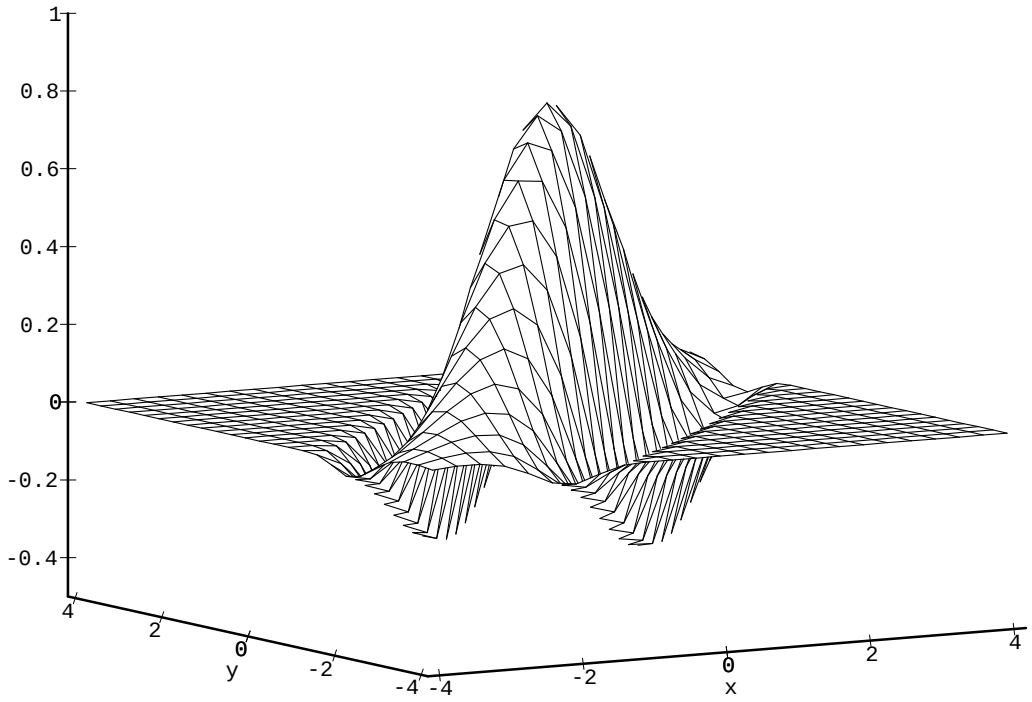

Figure 2: Representation of the filter $\mathbf{M_{\theta_1}}$.

This filtering is used in order to give a weight $\mathbf{W_{\theta_k}}$ to each direction $\theta_\mathbf{k}$.

$$W_{\theta_k}(s) \;=\; \sum_{r \in \mathcal{V}_s} I(r)\;\; M_{\theta_k}(s-r) \qquad (1)$$

$\mathcal{V}_s$ = 8-neighborhood of $\mathbf{s}$.

Then, after a normalization of the $\mathbf{W_{\theta_k}(s)}$, we obtain easily, for a site $\mathbf{s}$, the "best" direction $\theta_\mathbf{max}$ given by :

$$\theta_{max}(s) \qquad \text{such that} \qquad W_{\theta_{max}}(s) \;=\; \max_k \;(W_{\theta_k}(s)) \qquad (2)$$

3.1.2 Second step : Creation of contour chains.

The second step of this initialization allows the determination of contour chains, given the "best" direction.
The contour chain is made up of a set of connected pixels approximately aligned. A chain is, in fact, a set of connected pixels that have the same privileged direction.

Given a chain **ch**,

$$\forall\ s, r \text{ neighbor pixels} \quad : \quad (s \in ch\ et\ r \in ch) \quad \Leftrightarrow \quad (\theta_{max}(s) \ = \ \theta_{max}(r)) \qquad (3)$$

Scanning the image allows the determination of the different chains.

3.1.3 Third step : Suppression of small chains.

Small isolated chains correspond often to noise. So in order to correctly use the mean direction of the chain, which is the direction in which the contour can be prolonged, the chain length must be reasonably large.

For this reason we process small chains, in order to either delete them if they are isolated or insert them into longer adjacent chains.

Those three steps allow the subdivision of contours into chains, so we can then initialize the first component of the MRF : the chain number.

We will now describe the initialization of the two other components.

3.2 Initialization of the direction.

The second component of the **MRF**, the direction, has been already defined as a vector $\mathbf{D}_\theta(\mathbf{s})$ with four components ($\mathbf{D}_{\theta_\mathbf{k}}(\mathbf{s})$ ($\mathbf{k} \in [0..3]$)).

This vector enables us to have, at each site, information about the mean direction of the contour line, which is the direction in which the contour is supposed to be prolonged.

All the pixels of a chain are initialized with the same vector $\mathbf{D}_\theta(\mathbf{s})$.

In fact, for each chain, the value of $\mathbf{D_{\theta_k}(s)}$ determines a mean measure (average on the set of pixels belonging to the chain) of the likelihood of the direction $\theta_\mathbf{k}$.

To initialize the direction, we start by determining for every site, a local measure $\mathbf{d_{\theta_k}(s)}$ of the likelihood of the direction $\theta_\mathbf{k}$.

Some notation :

$\mathcal{V}_{s,\theta_k}$ is the set of the neighboring sites of **s** in the direction $\theta_\mathbf{k}$.

Or more formally :

$$\mathcal{V}_{s,\theta_k} = \{\ r \in \mathcal{V}_s \text{ such that } (r,s) \text{ makes an angle of } \theta_k \text{ with the horizontal}\} \quad (4)$$

We define the value $\mathbf{d_{\theta_k}(s)}$ as the sum of the norm of the gradient, on all the neighbors **r** (with $r \in \mathcal{V}_{s,\theta_k}$) :

$$d_{\theta_k}(s) = \sum_{r \in \mathcal{V}_{s,\theta_k}} I(r) \quad (5)$$

To determine the direction of a chain **ch**, we compute the average of the local values d_{θ_k}, on the set of sites **s** belonging to **ch**.

$$D_{\theta_k}(s) = \frac{\sum_{s \in ch} d_{\theta_k}(s)}{N_s} \quad (6)$$

where N_s is the number of the sites belonging to **ch**.

After normalization :

$$D_{\theta_k}(s) = \frac{D_{\theta_k}(s)}{\sum_{k=0}^{3} D_{\theta_k}(s)} \quad (7)$$

For every pixel that belongs to the chain ch, the second component of the Markov Random Field is then initialized with the vector $\mathbf{D_{\theta_k}(s)}$ (where k $\in$ [0..3]).

3.3 Initialization of the state.

There are four possible states (see §2.3) :

- **No contour : O**
- **Termination : T**
- **Straight : D**
- **Corner : C**

To initialize the state of a pixel where a contour has been detected (else take state **O**), we first determine the number of different chains in its neighborhood (written **nb_ch**).

Next, we initialize according to the following configuration rules :

- "**C**" : If $(nb\_ch > 1)$
  $\longrightarrow$ **Intersection of different chains.**
- "**T**" : If $(nb\_ch = 1)$
  and ($(\exists r \in \mathcal{V}(s)$ *with state* $r \neq O)$
  or $(\exists\ r_1$ *and* $r_2 \in \mathcal{V}(s)$, $((r_1, \widehat{s, r_2})$ *acute), states* r_1 *and* $r_2 \neq O)$

  For example : **T D** **T D**
  D
  $\longrightarrow$ **end of chain : one or two neighbors detected**
  (See the example given below.)
- "**D**" : Else
  $\longrightarrow$ **Inside a chain.**

Example :

Initialization of states when there are two adjacent chains :

T D D C
C D
D D
T

4 Energy function.

The energy function is defined in order to minimize some criterion. As we want to reconstruct an image of incomplete contours, the minimal energy must be reached when the contours are restored. Thus, constraints must be selected to extend the contour lines in the "best" directions, and to create angles (and T-junctions), in order to obtain closed contours.

The total energy **U** is the sum, over all cliques, of potential functions.

$$U = \sum_{c \in \mathcal{C}} V_c \tag{8}$$

where :

- **c** represent a clique, and **C** the set of all cliques.
- $\mathbf{V_c}$: the clique potential function.

First, in order to define those potential functions, we have to introduce a few notations.

4.1 Notations.

- **D** is a boolean which relates to the initial data :

$$D(s) = \begin{cases} 1 & \text{If I(s)} \neq 0 \\ 0 & \text{Otherwise} \end{cases}$$

- $\mathbf{E_p}$ indicates the contour width :

$$E_p(s) = \begin{cases} 1 & \text{If the width of contour is greater than 1.} \\ 0 & \text{Otherwise} \end{cases}$$

Convention : Representation of a thick contour ($E_p(s) = 1$) :

• • **X** **X** **X** **X** • •

X : thick contour
• : thin contour

- **Br** is also a boolean which is taken into account in the energy function. This value enables us to have a high energy for the configuration which extends contours, in the case of a "corner" or of a "T-junction".

An example of the value of the boolean Br is given next page.